

FIG. 1

```
void PPS_fun()  
{  
    i = 0; 282  
    //the PPS loop  
    for (;;) 284  
    {  
        u(i);  
        i = d(); 286  
    }  
}
```

280

FIG. 2A

```
void PPS_fun_thread_0()  
{  
    for (;;) 302  
    {  
        wait for thread 1  
        u(i);  
        i = d();  
        inform thread 1 304  
    }  
}
```

300-1

FIG. 2B

```
void PPS_fun_thread_1()  
{  
    i = 0; 316  
    inform thread 0 318  
    for (;;) 312  
    {  
        wait for thread 0  
        u(i);  
        i = d();  
        inform thread 0 314  
    }  
}
```

300-2

FIG. 2C

```
Void PPS_fun_thread_0()  
{  
  for (;;)   
  {  
    AWAIT(n) ~ 322  
    u(i);  
    i = d();  
    ADVANCE(n) ~ 324  
  }  
}
```

300-1

FIG. 3A

```
Void PPS_fun_thread_1()  
{  
  i = 0; ~ 326  
  ADVANCE(n) ~ 328  
  for (;;)   
  {  
    AWAIT(n) ~ 332  
    u(i);  
    i = d();  
    ADVANCE(n) ~ 334  
  }  
}
```

300-2

FIG. 3B

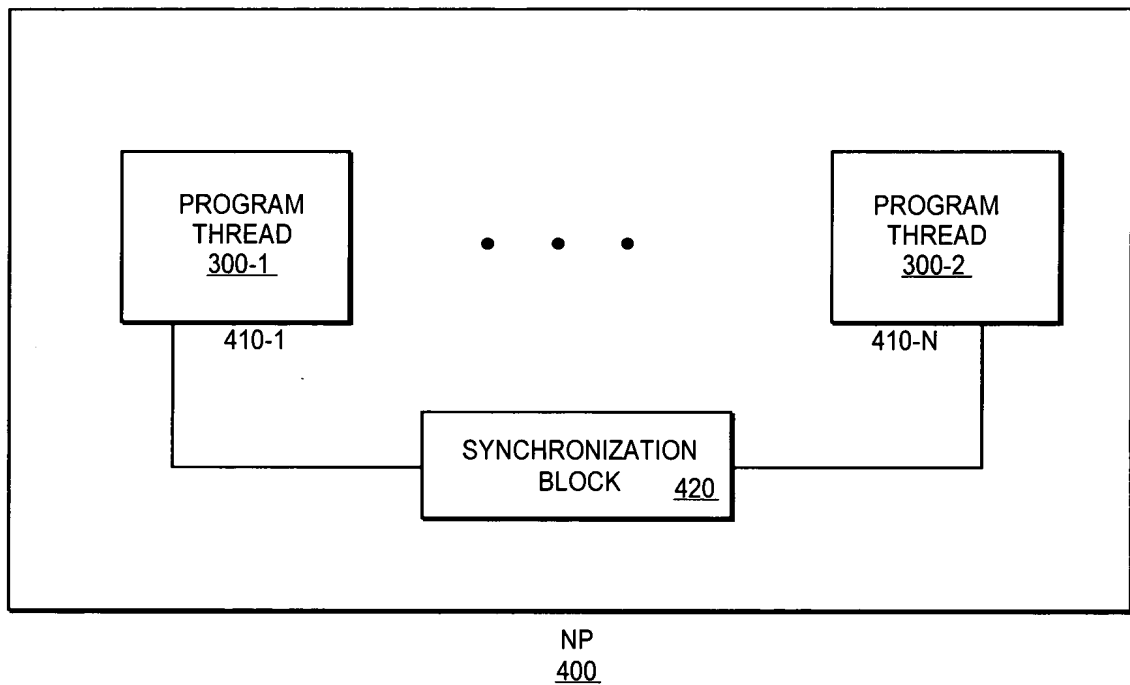


FIG. 4

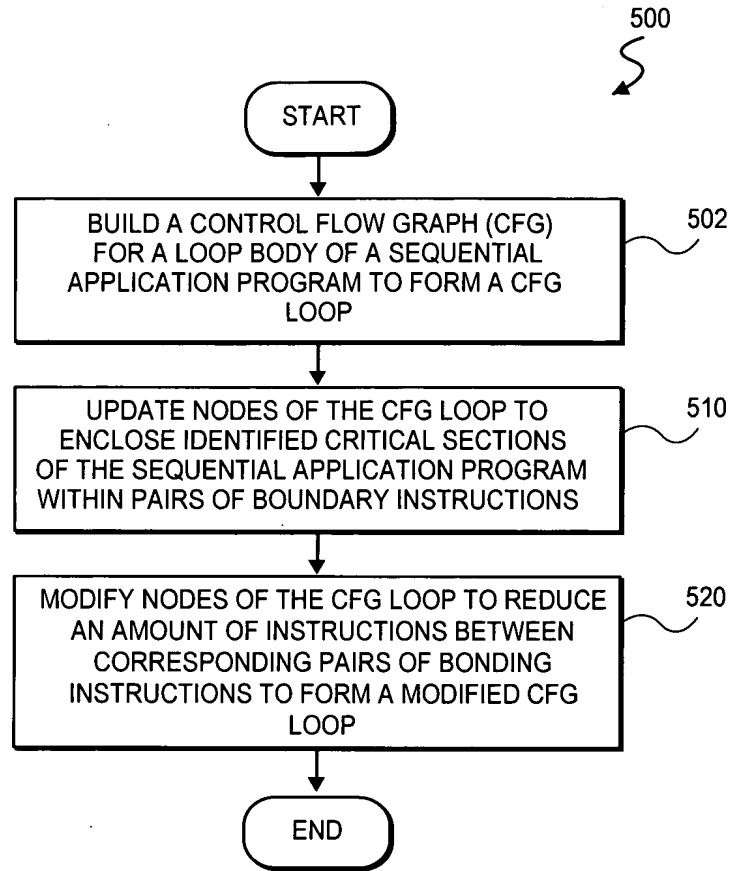


FIG. 5

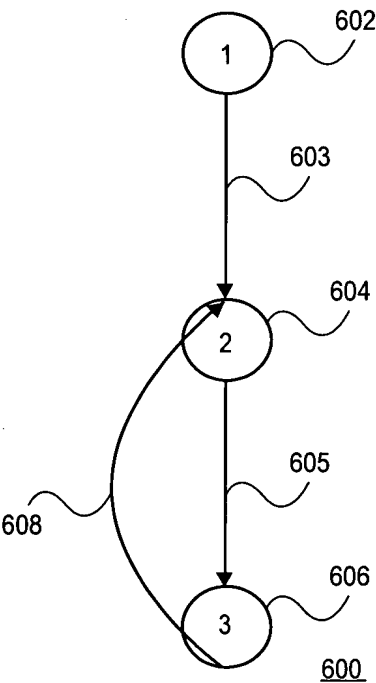


FIG. 6A

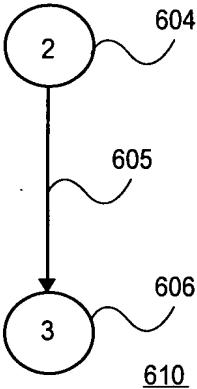
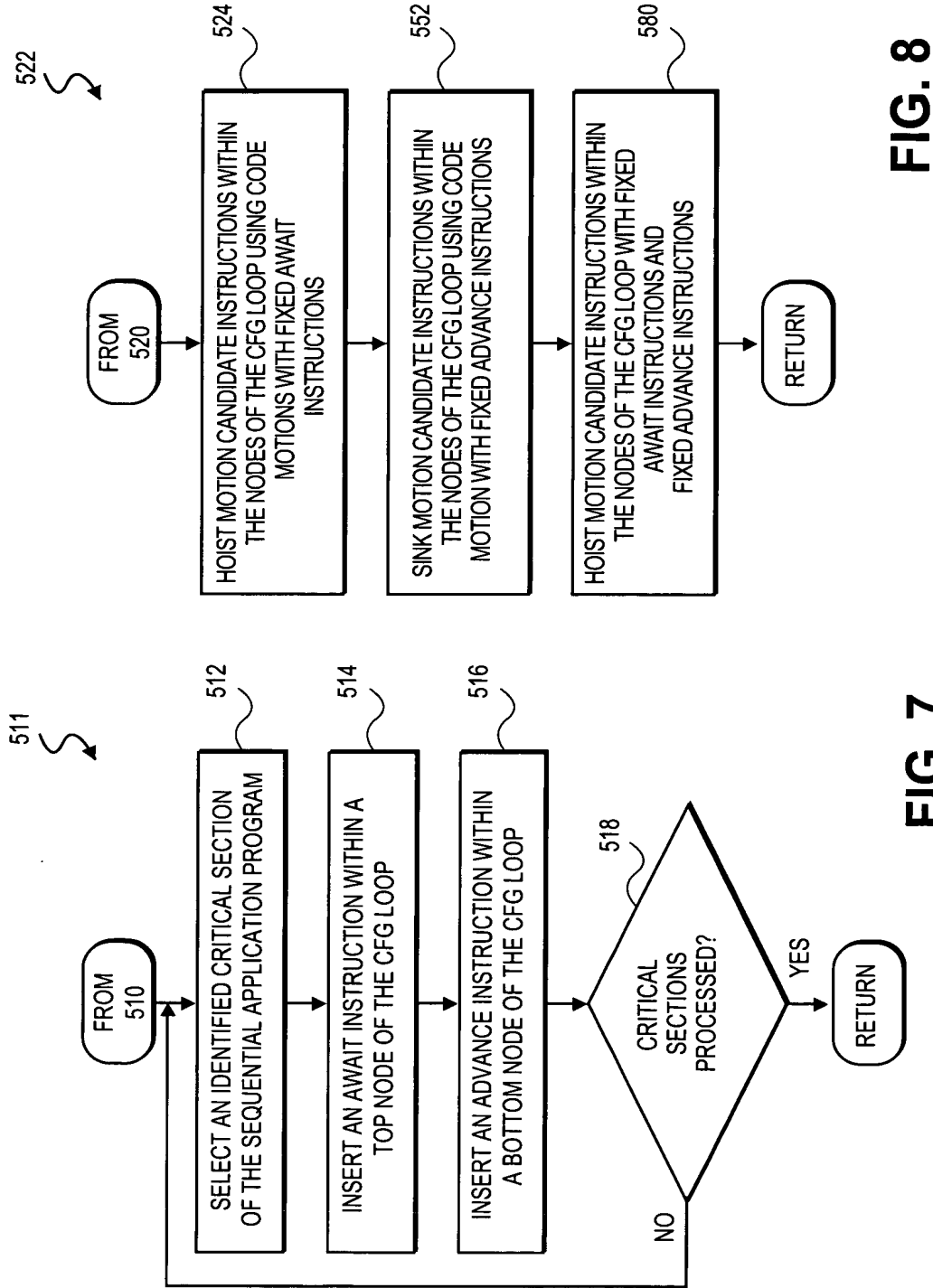


FIG. 6B



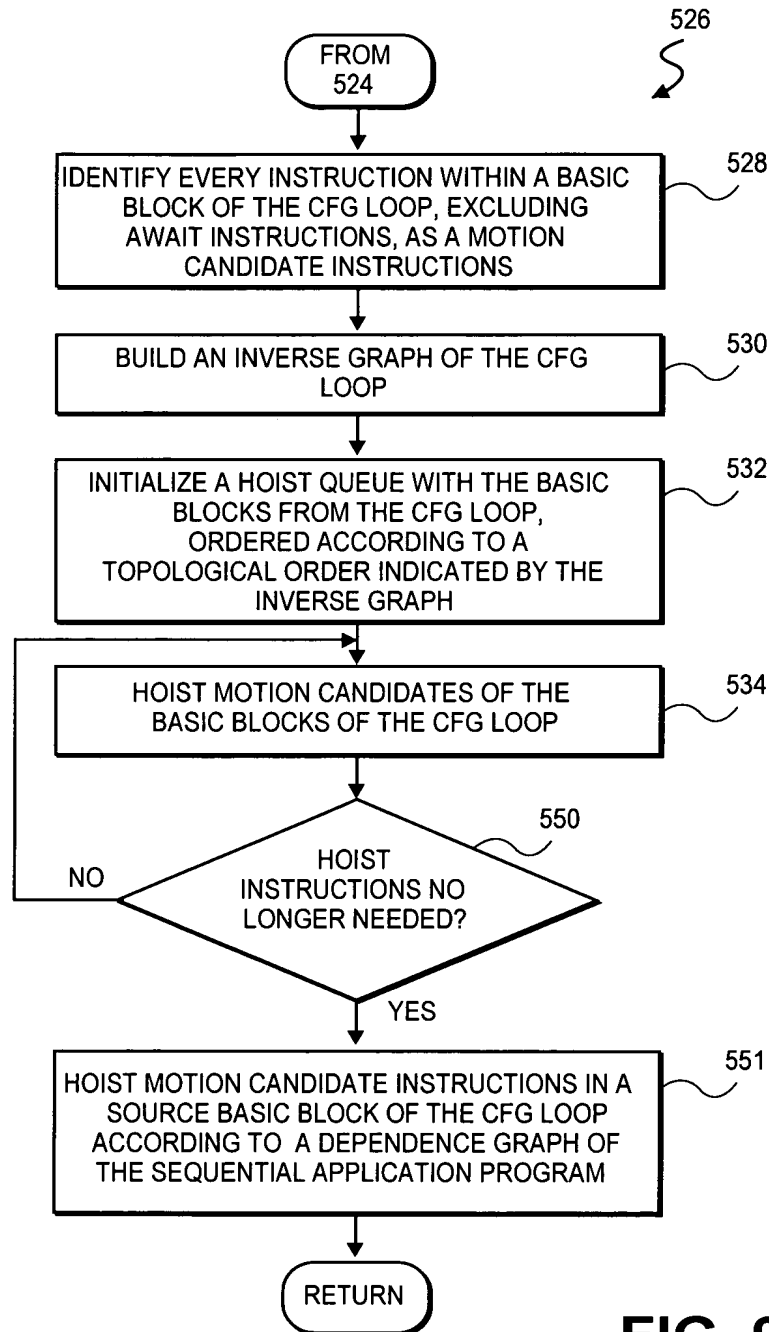
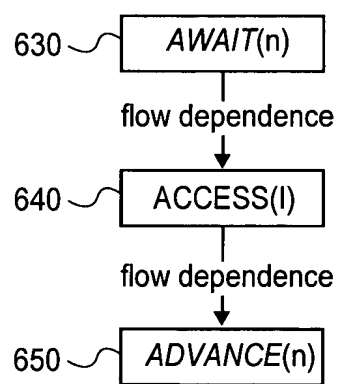


FIG. 9



620

FIG. 10

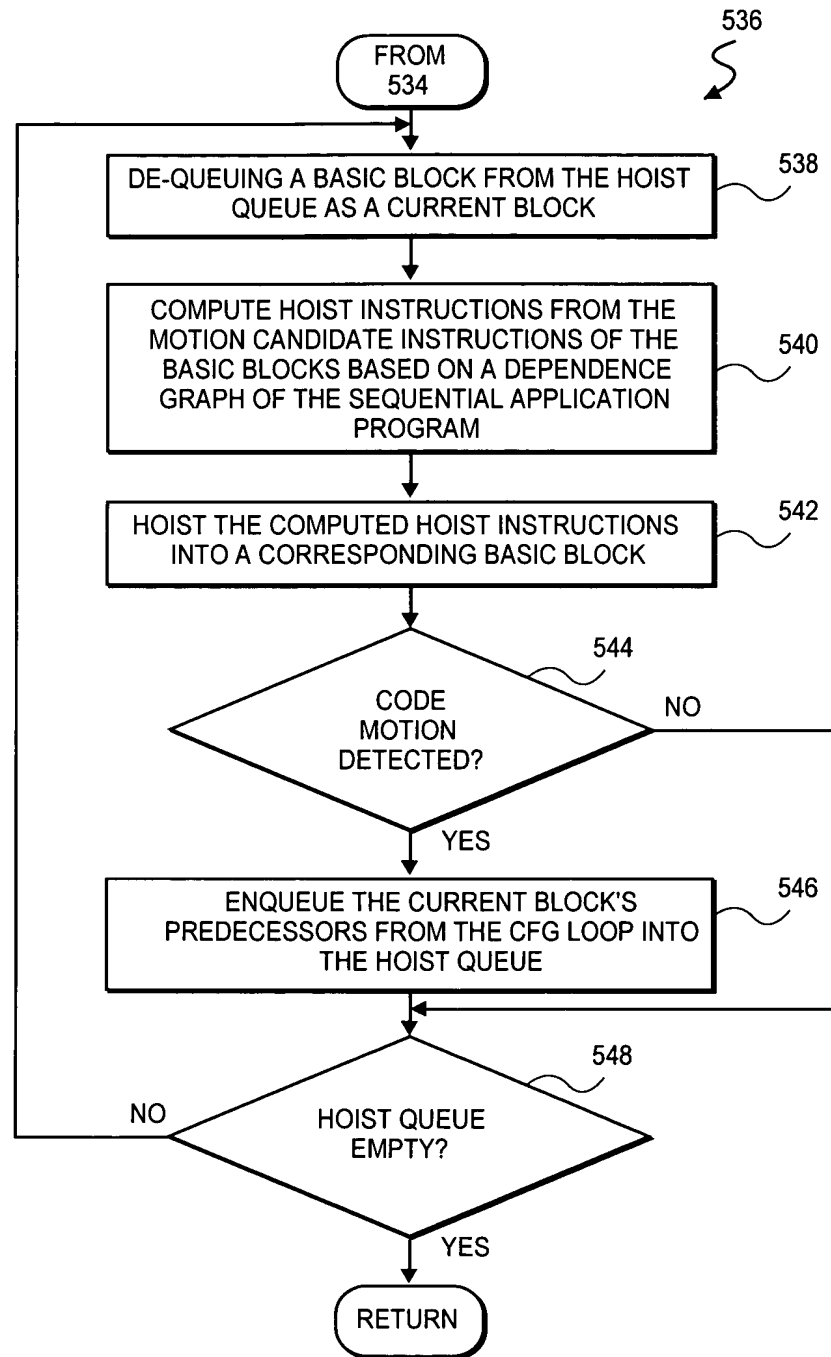


FIG. 11

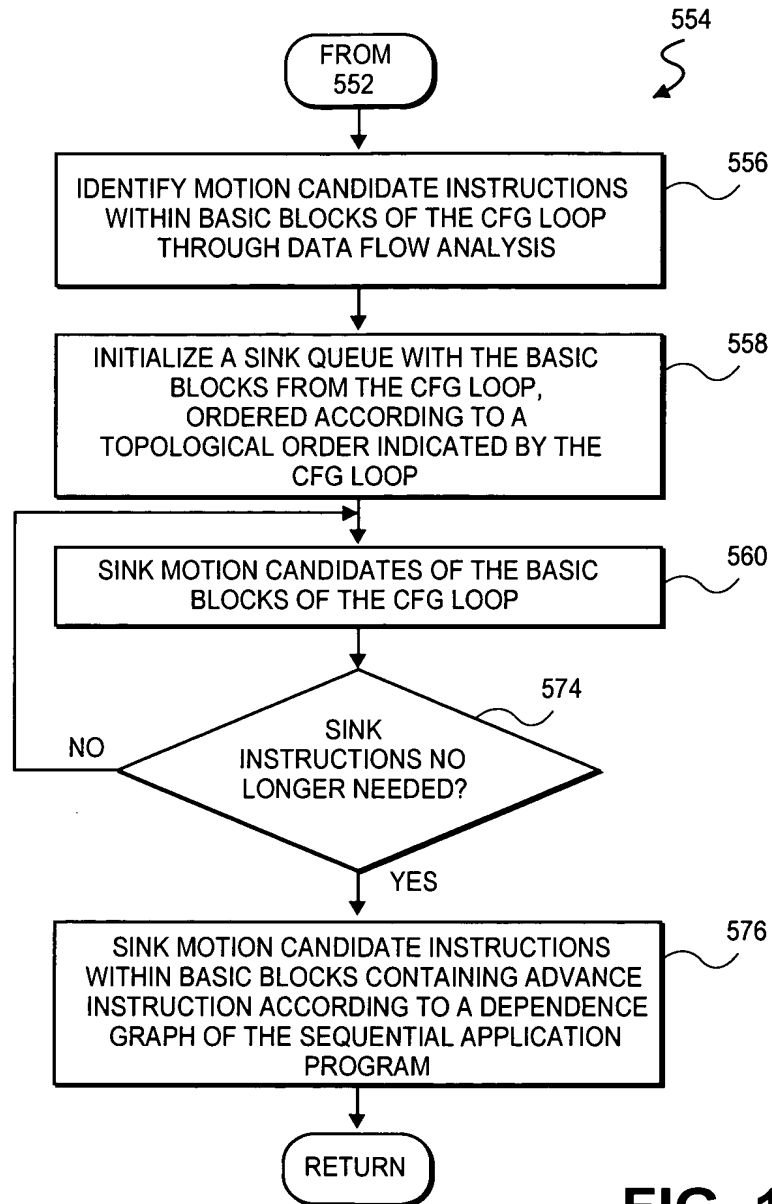
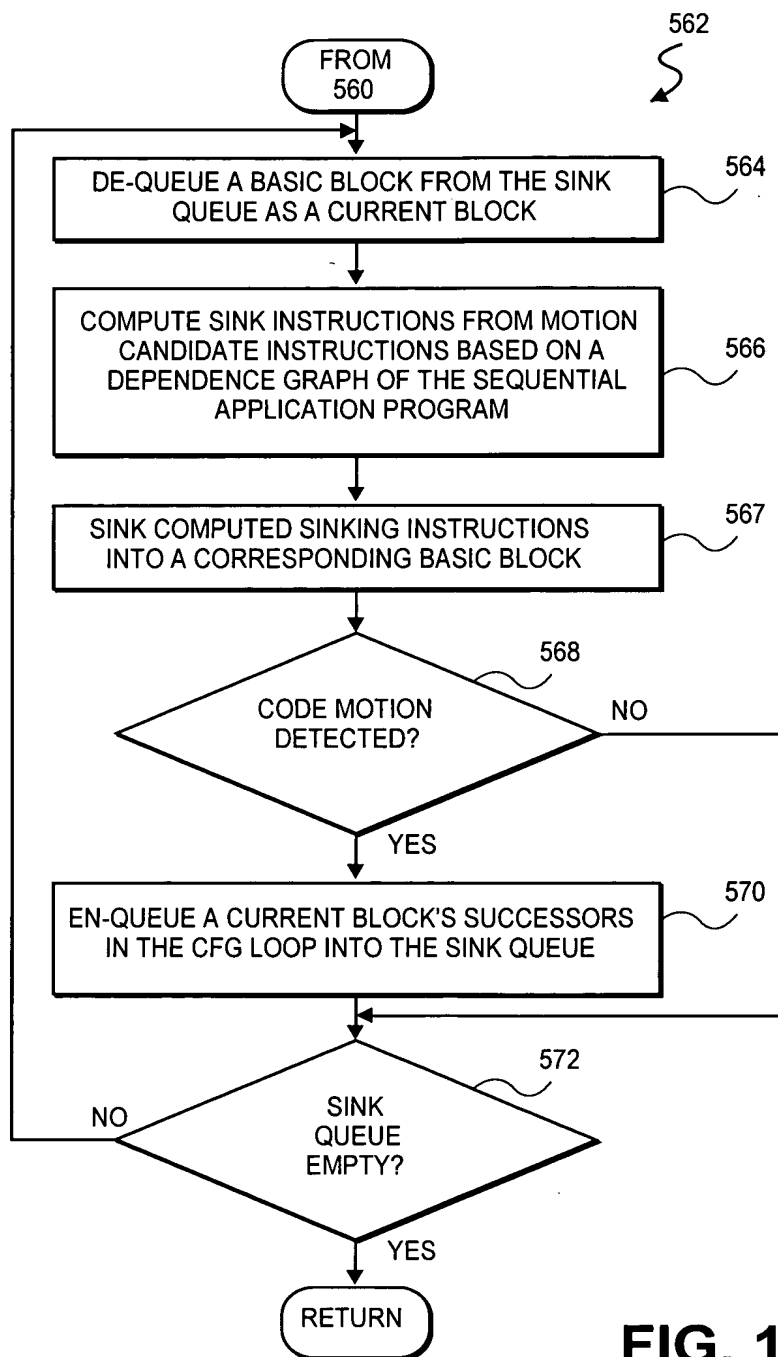
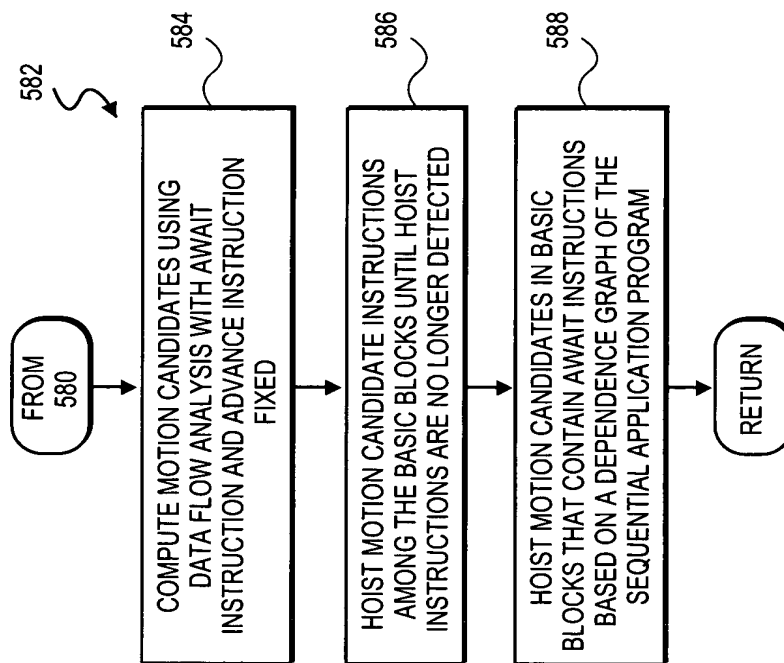
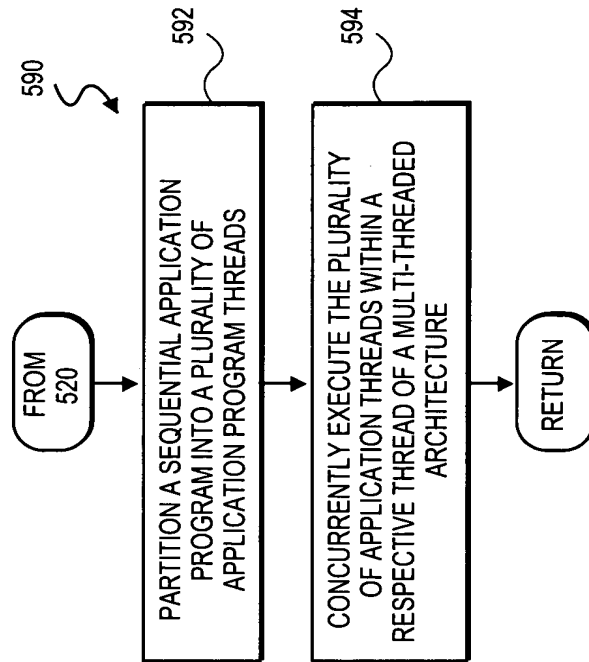


FIG. 12





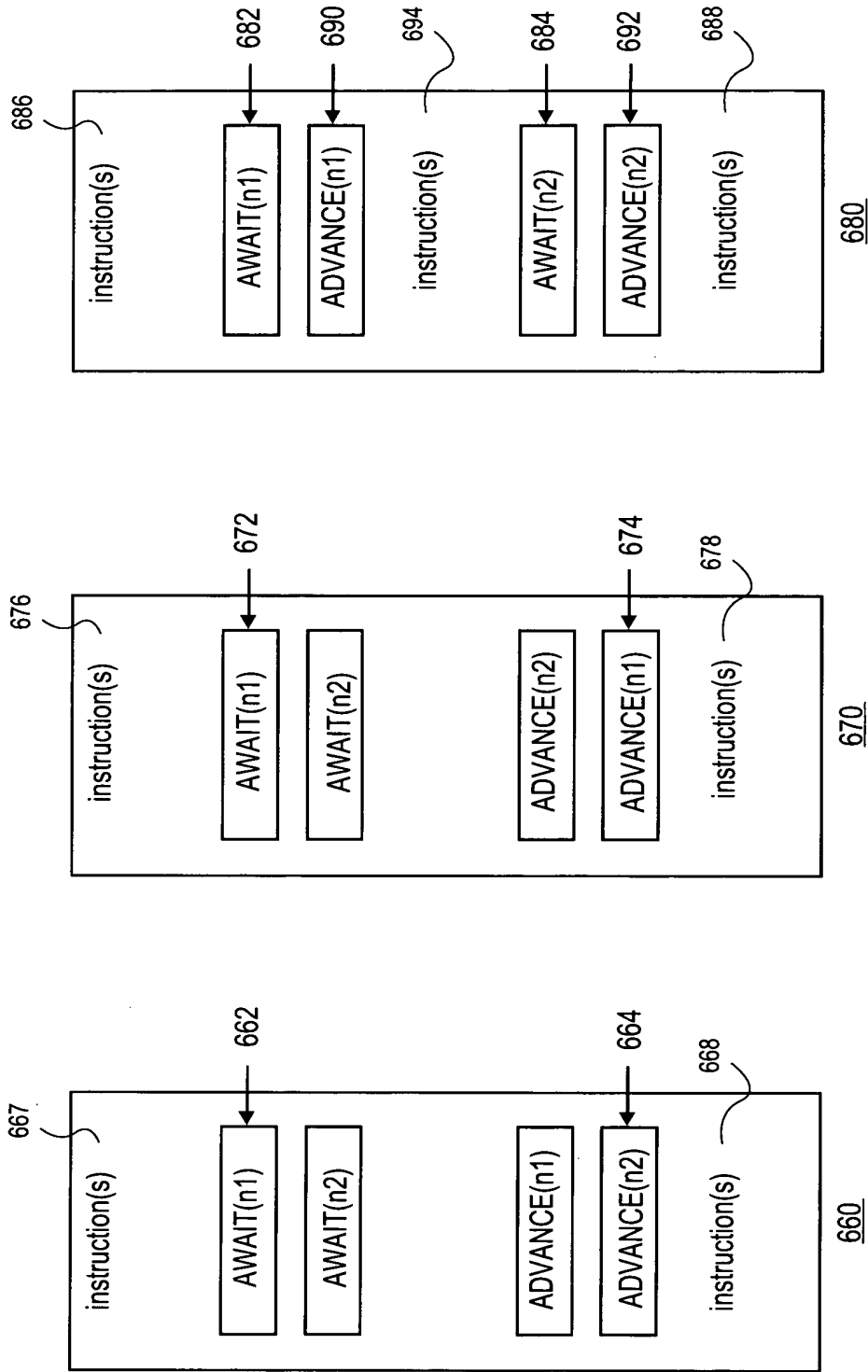


FIG. 15A

FIG. 15B

FIG. 15C